Supplement 1:

《Logic BERT》

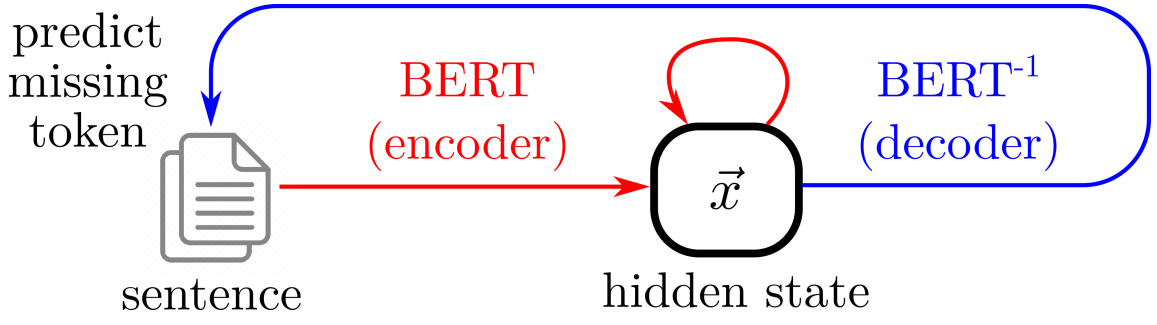
YKY

July 31, 2020

Table of contents

BERT 的革命性意义：「闭环路训练」

- BERT 利用平常的文本 induce 出知识，而这 representation 具有 通用性 (universality)：



(1)

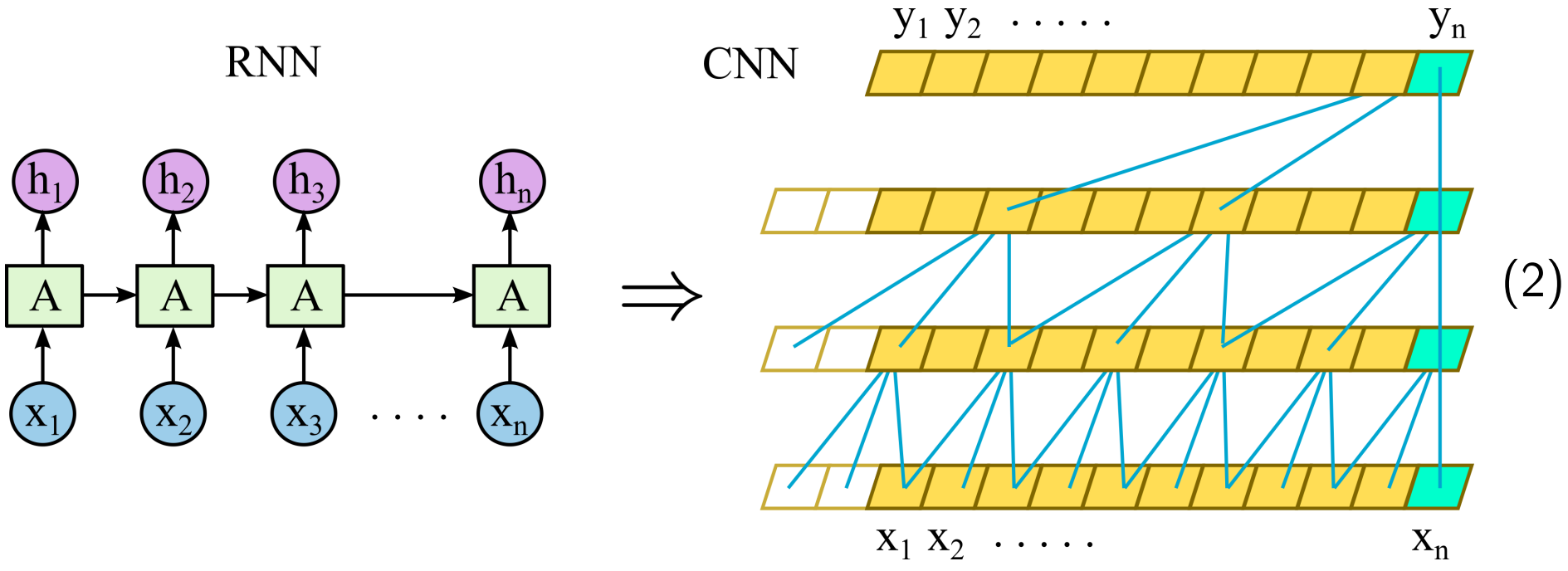
换句话说：隐状态的 representation 压缩了句子的意思，而它可以应用在别的场景下

- This implies that human-level AI can be *induced* from existing corpora, 而不需重复 像人类婴儿成长的学习阶段
- Such corpora can include items such as images, movies with dialogues / subtitles
- 这种训练方法是较早的另一篇论文提出，它并不属於 BERT 的内部结构

BERT 的内部结构

其实，BERT 也是混合了很多技巧 发展而成的：

- BERT 基本上是一个 seq-to-seq 的运算过程
- Seq-to-seq 问题最初是用 RNN 解决的
- 但 RNN 速度较慢，有人提出用 CNN 取代：



(2)

- CNN 加上 attention mechanism 变成 Transformer
- 我的目标是重复这个思路，但引入 symmetric NN 的结构

Symmetry in logic

- 词语 组成 句子，类比於 逻辑中，概念 组成 逻辑命题
- 抽象地说，逻辑语言 可以看成是一种有 2 个运算的 代数结构，可以看成是 加法 $\wedge$ 和 乘法 $\cdot$，其中 乘法 是不可交换的，但加法 可交换
- 例如：

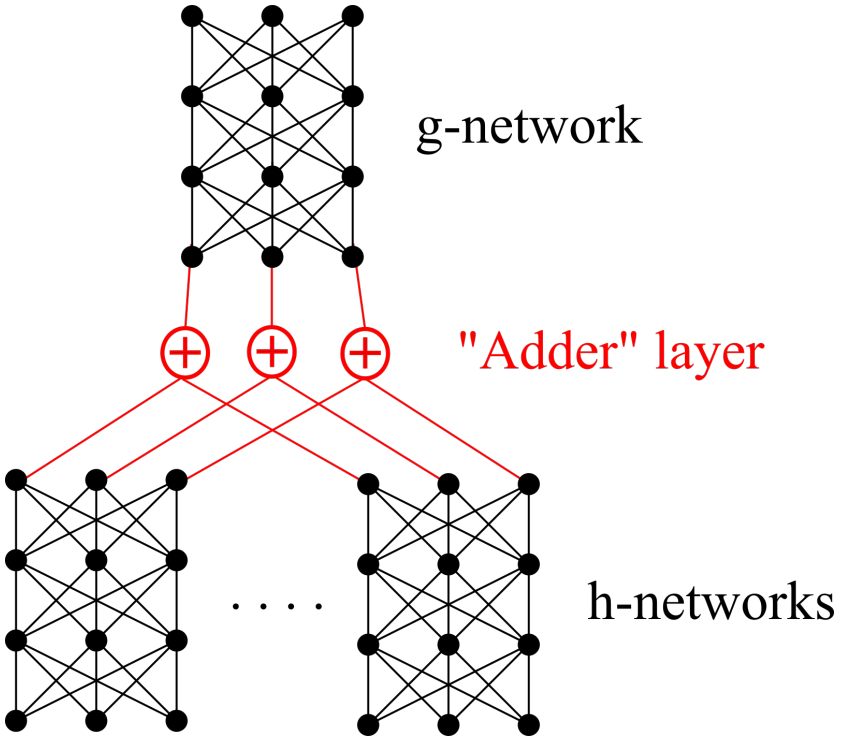
$$\begin{aligned} A \wedge B &\equiv B \wedge A \\ \text{下雨} \wedge \text{失恋} &\equiv \text{失恋} \wedge \text{下雨} \end{aligned} \tag{3}$$

- Word2Vec 也是革命性的；由 Word2Vec 演变成 Sentence2Vec 则比较容易，基本上只是 向量的 延长 (concatenation)；逻辑命题 类似於 sentence
- 假设 全体逻辑命题的空间是 $\mathbb{P}$，则 命题集合 的空间是 $2^{\mathbb{P}}$，非常庞大
- 如果限制 状态 $\vec{x}$ = working memory 只有 10 个命题，$\vec{x}$ 的空间是 $\mathbb{P}^{10}/\sim$ 其中 $\sim$ 是对称群 $\mathfrak{S}_{10}$ 的等价关系。换句话说 $2^{\mathbb{P}} \cong \coprod_{n=0}^{\infty} \mathbb{P}^n/\mathfrak{S}_n$
- $\mathbb{P}^n/\mathfrak{S}_n$ 虽然是 $\mathbb{P}^n$ 的商空间，但 $\mathfrak{S}_n$-不变性 很难用神经网络实现

Symmetric neural network

- Permutation invariance can be handled by symmetric neural networks
- 我浪费了两年时间试图解决这问题，却发现在 3 年前已经有两篇论文解决了 [PointNet 2017] [DeepSets 2017]，而且数学水平比我高很多
- Any symmetric function can be represented by the following form (a special case of the Kolmogorov-Arnold representation of functions):

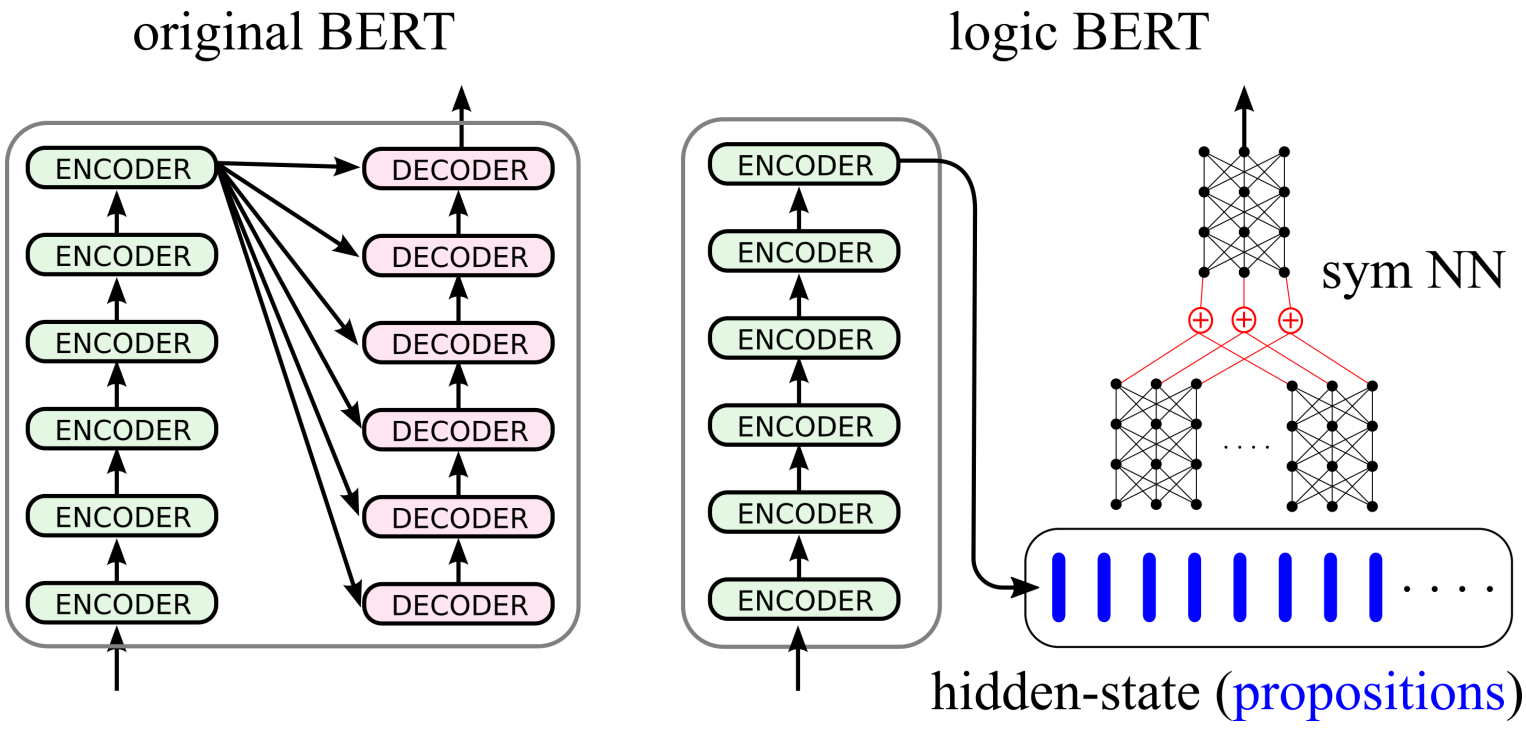
$$f(x, y, ...) = g(h(x) + h(y) + ...) \tag{4}$$



(5)

BERT 的逻辑化

- 可以将 BERT 的 隐状态 变成 “set of propositions” 的形式，方法是将 原来的 decoder 变成 sym NN：



(6)

- 这时，输出对 | 的交换不变性 is automatically satisfied by the architecture of the decoder
- 原来的 encoder 可以照旧使用，，因为它是一个 universal seq-2-seq mapping
- 因为后半部改变了，error propagation 会令 representation 也改变
- 当然，这个想法有待实验证实 ☺

Attention 是什么?

- 注意力 最初起源於 Seq2seq，后来 BERT 引入 self-attention
- 在 Seq2seq 中，编码器 (encoder) 由下式给出，它将输入的词语 x_i 转化成一连串的 隐状态 h_i:

$$h_t = \text{RNN}_{encode}(x_t, h_{t-1}) \tag{7}$$

- 这些 h_i 可以综合成单一个 隐状态 $c = q(h_1, ..., h_n)$.
- 这个 c 被「寄予厚望」，它浓缩了整个句子的意义
- 解码器 的结构类似，它的隐状态是 s_t，输出 y_t:

$$s_t = \text{RNN}_{decode}(y_t, s_{t-1}, c_t) \tag{8}$$

- 注意最后的 c_t 依赖时间，它是隐状态 h_j 的 加权平均:

$$c_i = \sum_j \alpha_{ij} h_j \tag{9}$$

- 其中 α_{ij} 量度 输入 / 输出 的隐状态之间的 相似度，取其最大值:

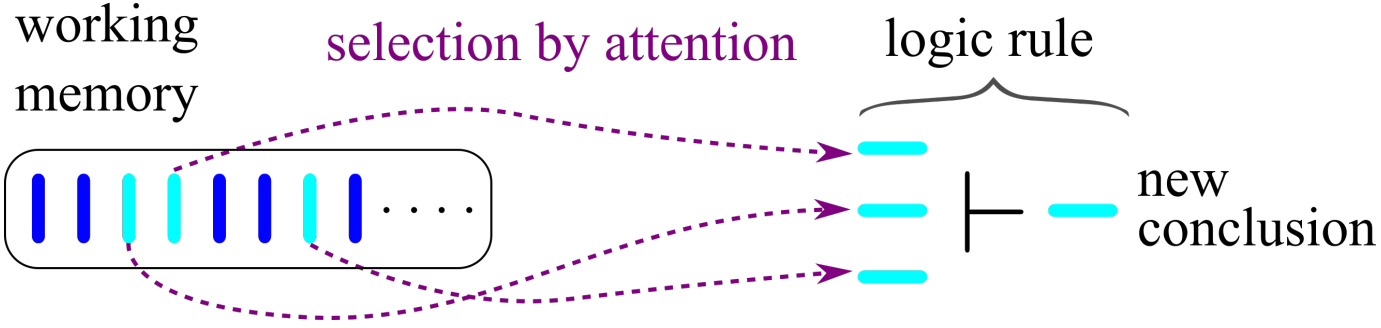
$$\alpha_{ij} = \text{softmax}\{\langle s_i, h_j \rangle\} \tag{10}$$

换句话说，α_{ij} 选择 最接近 h_j 的 s_i

Attention 给逻辑 AI 的启发

我这样理解 attention：

- 例如，翻译时，输入／输出句子中「动词」的位置可以是不同的
- 当 解码器需要一个「动词」时，它的隐状态 s_t 含有「动词」的意思
- Attention 机制 找出最接近「动词」的 编码器的隐状态（可以 ≥ 1 个）$\sum h_j$，交给 解码器，这是一种 information retrieval
- 例如，将 M 件东西 映射 到 N 件东西，可以有 N^M 个 mappings，这是非常庞大的空间。但如果这些物件有 类别，而同类只映射到同类，则可以用 attention 简化 mappings
- 所以 attention 是一种 inductive bias，它大大地缩小 mapping 空间
- 在逻辑的场景下，需要的 mapping 是 f : 命题集合 $\to$ 命题



(11)

“Attention is all you need” ?

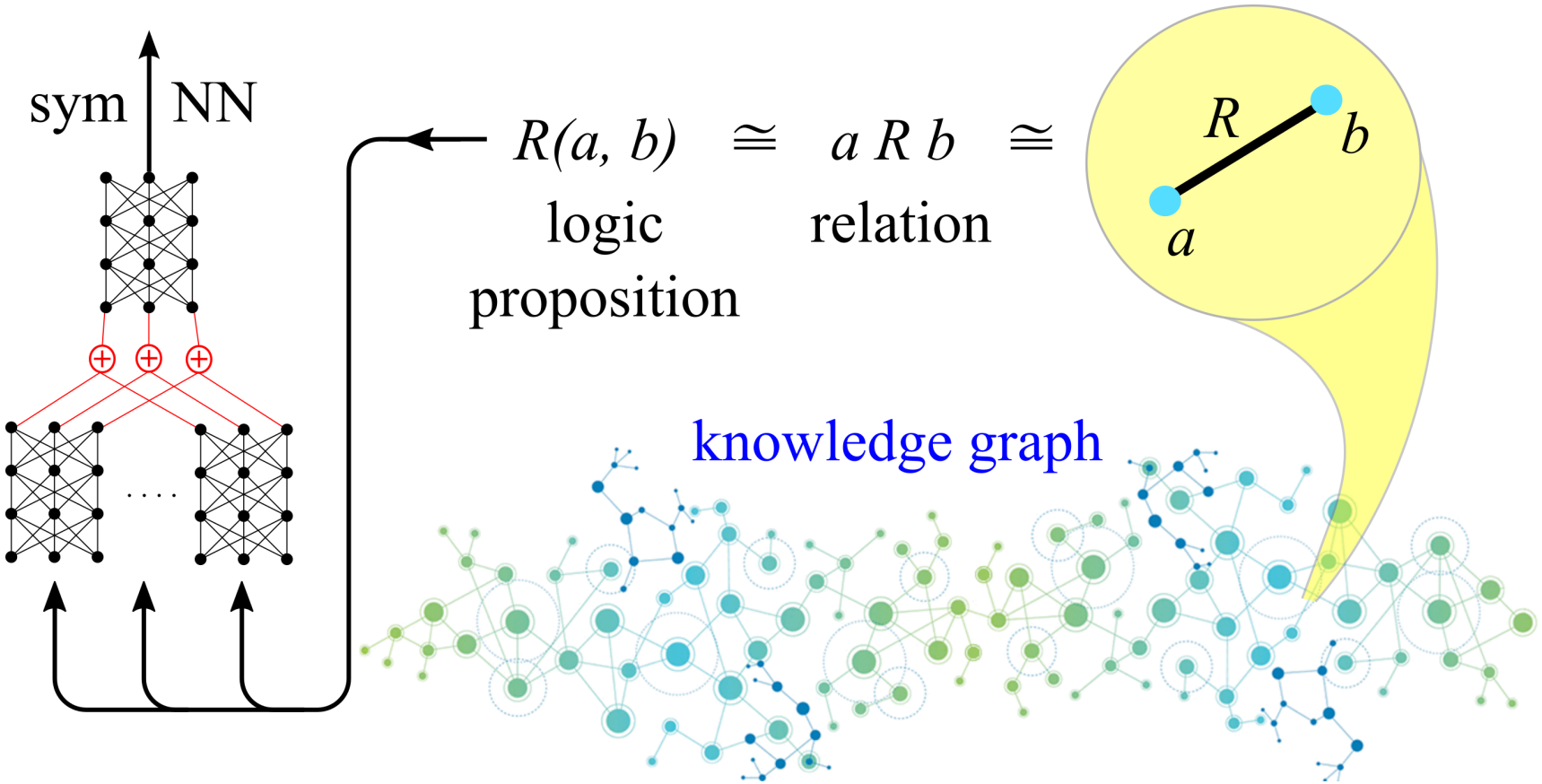
- 但逻辑 attention 和一般 attention 要求略有不同
- 不是「同类映射到同类」，而是要在庞大的 logic rules 空间中找到适用 (applicable) 的 rules
- 隐状态 s_t 代表 “search state”，注意力 的目的是 选择 s_t 所需要的那些命题，交给 解码器
- 注意：逻辑 attention 从 M 个命题中 选择 N 个命题，$M > N$. 这是 inductive bias. 而 symmetric NN 的做法，只是要求 M 个命题 的 置换不变性，所以它浪费了资源在很多 “don’t care” 的命题上
- 换句话说，attention 或者可以加强 sym NN 的效率，甚至取代 sym NN
- 假设 q 是（包含多个命题的）逻辑 state，$\bowtie$ 表示 matching by attention:

$$q \bowtie \text{rule} \to \text{conclusion} \tag{12}$$

Thus, each row in the attention matrix 对应於一条 logic rule. 这是一种 “flat” 的 representation of rules, 但 sym NN 采用的是有「深度」的 representation. 这是值得再思考的问题

应用：知识图谱 (knowledge graphs)

- 知识图谱 不能直接输入神经网络，它必需分拆成很多 edges，每个 edge 是一个 关系，也是一个 逻辑命题 ；也可以说 “graphs are isomorphic to logic”

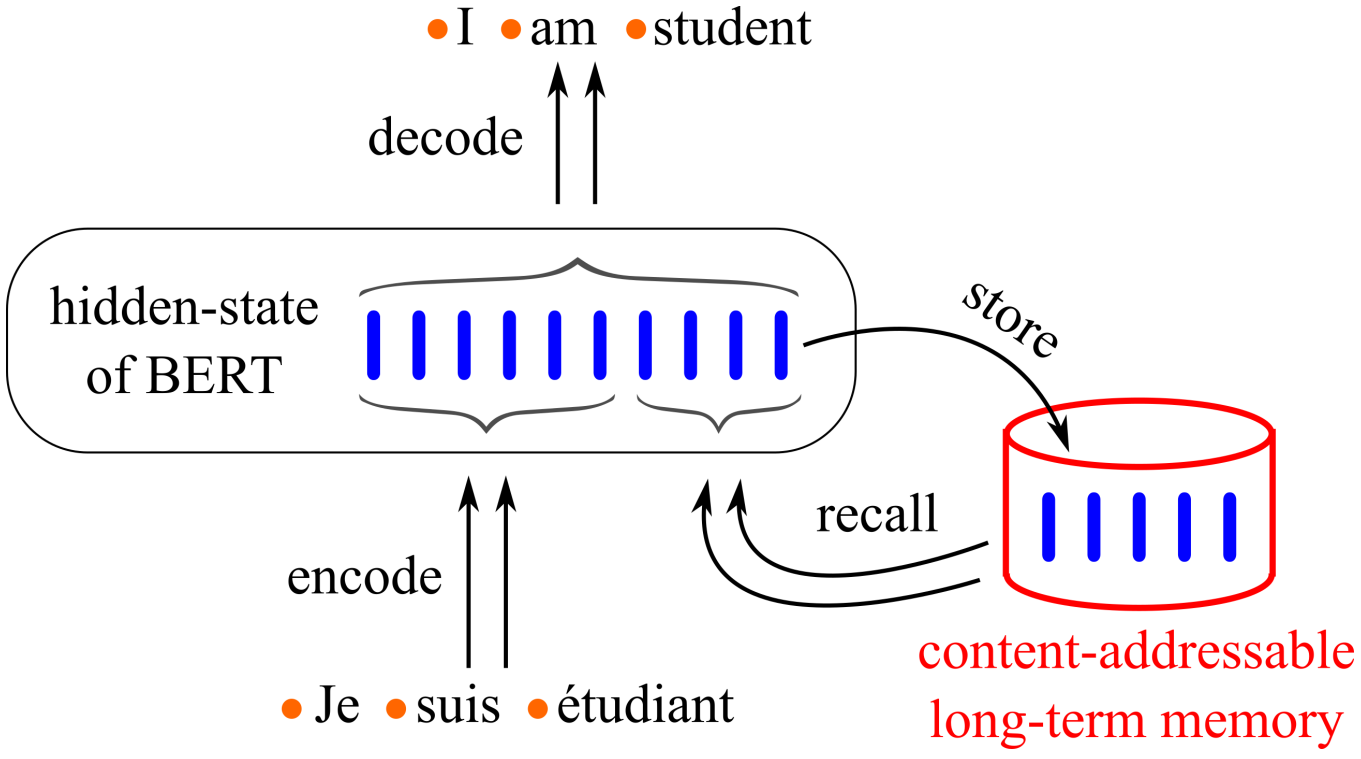


(13)

- 而这些 edges 似乎必需用 symmetric NN 处理，因为它们是 permutation invariant

应用：content-addressable long-term memory

- 以前 BERT 的隐状态 没有逻辑结构，我们不是很清楚它的内容是什么；逻辑化之后，BERT 内部的命题可以储存在 长期记忆 中：



(14)

- 这种系统 已非常接近 strong AI，而这是有赖 逻辑化 才能做到的
- Content-addressable memory 的想法来自 Alex Graves *et al* 的 Neural Turing Machine [2014]

References

[1] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *CoRR* abs/1410.5401 (2014). arXiv: 1410.5401. URL: http://arxiv.org/abs/1410.5401.

[2] Qi et al. "Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *CVPR* (2017). https://arxiv.org/abs/1612.00593.

[3] Zaheer et al. "Deep sets". In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 3391–3401.